

Digital grusväg 1/2004

Dg 1/2004 (2004-02-15). Aperiodisk tidskrift. Årgång 9. (c) Sixten Borg.

Syntetiska intryck och uttryck.

Det har blivit tradition att inleda varje nummer av Digital grusväg med lite reflektioner. Det här är tredje försöket. Den första handlade om höstrusk, den andra om julen som nalkades. Men det tar tid att skriva ihop själva innehållet, så jag har fått byta ut inledningen efter hand.

Det här numret om syntetiska uttrycksformer: simulerade kartor, en deterministisk men syntetisk ö, ett gammalt berg som veckades redan 1993, syntetisk musik och lite datorgenererad konst.

Jag tog med lite mer konst för ventilations-system också. Det var kul att bygga en lampa, så det var kul att få visa upp den. Min bild är tyvärr inte så bra, jag hade problem med mot-ljus när jag tog den.

Mvh
Redaktionen.

Konst för ventilationssystem, del 2.

[Se Dg 1/2003]. En bit ventilationsslang blev till en lampa, invändigt upplyst av en halogenlampa. Slangen är krökt, så att man inte kan se ljuskällan utifrån, bara skenet som reflekteras mot slangens insida. Färgen och stilen passade bra in i mitt hem, som består av grå putsade väggar.



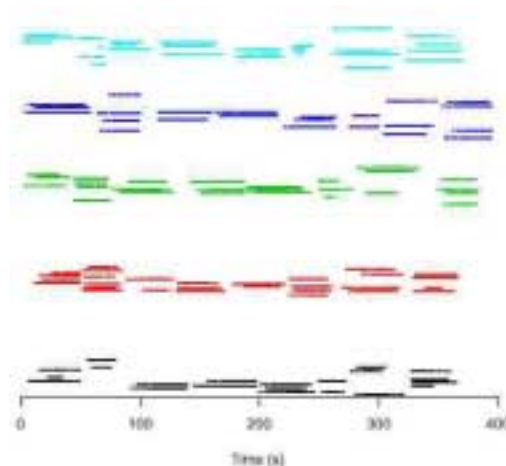
Digital grusväg: Aperiodisk tidskrift. Årgång 9. (c) Sixten Borg.
Korrespondens: elektronisk post digital.grusvag@swipnet.se.
Nätplats: <http://home.swipnet.se/digital-grusvag>
Digital grusväg är ett registrerat varumärke.

Manisk minimalism.

I förra numret skrev jag om musiksimulering. Detta man har inte gått över, och jag har gett ut en ny skiva. Marknaden mättades denna gång efter ett (1) exemplar, som jag i min välvilja behöll själv. Jag lät mig inspireras av den så kallade minimalismen. Ett reportage i en dagstidning (Sydsvenska dagbladet) tipsade om Philip Glass (redan en av mina favoriter) samt Steve Reich, som har en snygg hemsida och en gratis mp3-fil med ett helt underbart musikstycke, samt Steve Reilly som har en av internets absolut fulaste hemsidor. Jag vände direkt så jag vet inte hur han låter och inte ifall han bjuder på gratis musik. Sålunda inspirerad av Philip Glass och Steve Reich hackade jag ihop nya R-program och framställde ett antal stycken (R : Copyright 2003, The R Development Core Team, Version 1.6.2 (2003-01-10), <http://www.r-project.org>).

Min metod var att börja med en sekvens, t.ex. av sextondelstoner, som repeteras. Var och en av dessa har en "tröskel", hur lång tid det dröjer innan den byts ut mot motsvarande ton i en annan sekvens av sextondelar. Jag går under en tidsrymd gradvis över från att spela den ena sekvensen till den andra sekvensen.

Resultatet blir otroligt roligt om man är road av det. De två sekvenserna kan vara tagna från samma skala eller ackord, eller från olika. Man kan spela flerstämmigt, med fjärdedelar, åttondelar, sextondelar osv samtidigt. Första och sista sekvensen i varje stämma kan vara tom (ingenting spelas) så att stycket kommer smygande fram och avtar likartat. Etc. Vidstående diagram försöker illustrera ett musikstycke (jag behärskar ej notskrift).



Varje färg är en stämma, rektanglarna visar att en ton spelas. För respektive stämma visar den lodräta skalan frekvens.

Man kan naturligtvis mixtra med syntrösterna (ljuden). Jag genererar numera allting från summor av sinusfunktioner, för att få med övertoner. Varje röst är summan av grundtonen och ett antal övertoner, med olika styrkor. Beräkningsarbetet med att generera styckena var omfattande, och diverse knep krävdes. Jag skrev en modul i C för att snabba upp en del av programmet, vilket hjälpte mycket.

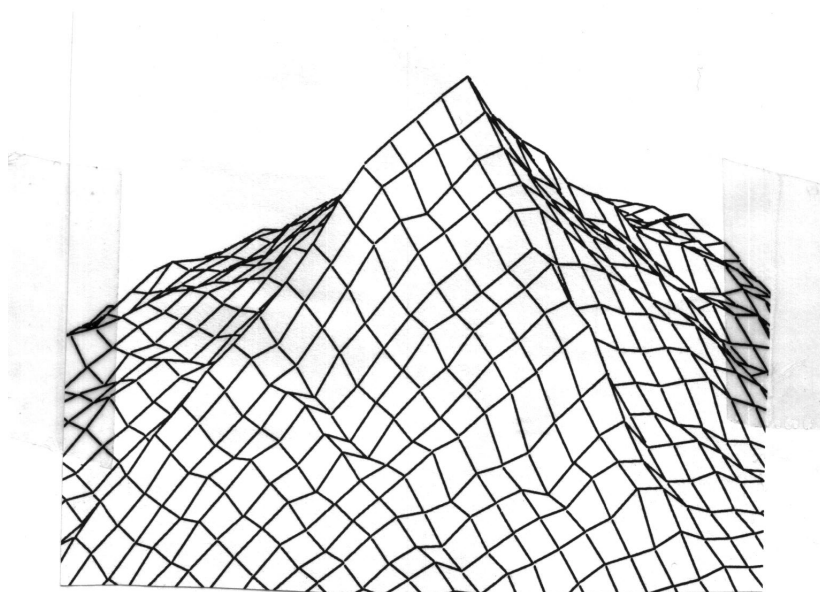
För att inte förpesta bandbredden totalt, nöjer jag mig med ett exempel från denna skiva: Klipp ur [Akustisk ärrvävnad \(1 minut, 550 kbyte\)](#).

[meddelande]

Redaktionen meddelar att man inte längre har möjlighet att ta emot och bearbeta alster på 8-tums och 5.25-tums disketter. Vi ber om överseende med detta.

Massivsimulering. Tillämpad regressionsanalys.

En gång var jag nästan lärare på en statistik kurs. Mitt jobb var att hjälpa eleverna med räkneövningar, kontrollera att de räknat rätt mm. Som ett glatt tillrop demonstrerade jag hur man kan tillämpa linjär regression, och simulera ett bergmassiv. Nedan ser ni hur det kan bli...



Tillämpning på linjär regression.

Ett berg byggs upp av att man delar upp en yta rekursivt och anpassar ett plan till varje del. Varje plan anpassas till de punkter som redan är beräknade.* Med hjälp av planet och slumpstal** skapas de punkter som krävs för att nästa uppdelning skall kunna göras, osv.

* MK-metoden

** Normalfördelade ($E[X]=0$, $V[X]=k \cdot (\text{planet's yta})$)

Simulerat berg. S Borg 1993. Återfunnet på ett papper i en gammal pärm.

Kvinnor har fler bröder än män med samma genetiska bakgrund.

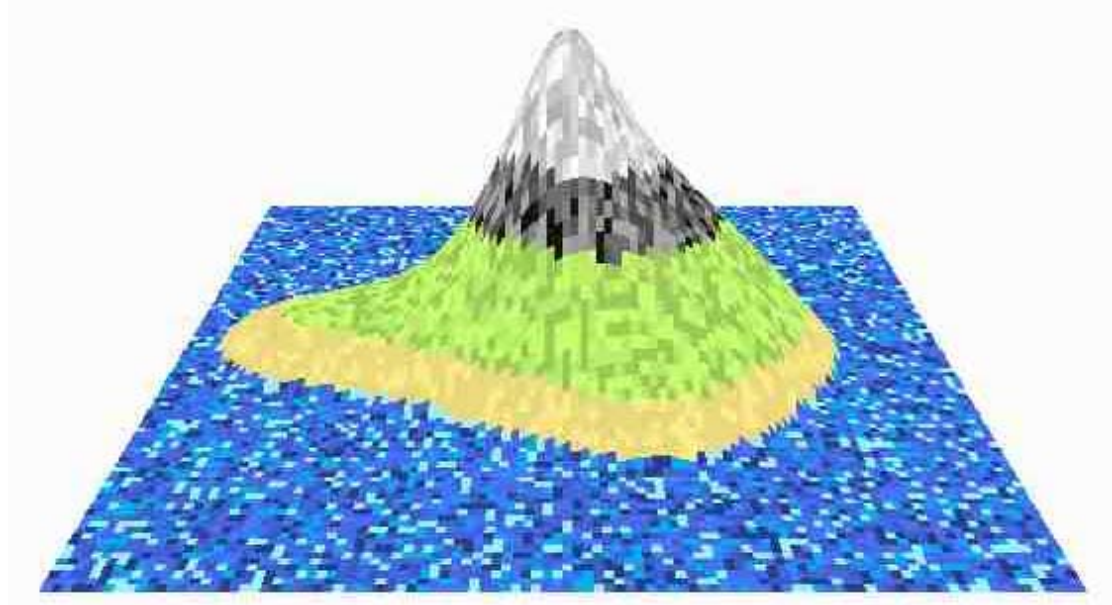
Jag och min stackars bror har bara en bror var, men våra systrar har turen att ha två bröder var. Dessutom är de fler än vi. Ett vägt genomsnitt är 1.6 bror per syskon. I en slumpmässigt vald kontrollgrupp råder en ännu mer särpräglad situation. Där har gruppen av kvinnor i genomsnitt en bror, medan gruppen av män i genomsnitt saknar bröder helt. Gruppen består av min vän U och hans syster.

Denna orättvisa gav upphov till en grävning i statistik årsbok. En montecarlosimulering av atomliknande kärnfamiljer (noll eller flera barn med samma föräldrar till samtliga familjens barn) med 1.547 barn per moder, samt av de födda barnen, 52% pojkar och 48% flickor (Uppgifter från: Befolkningsstatistik i Sverige år 2000; från Statistisk Årsbok 2003, SCB), visar att män har i genomsnitt 0.28 färre bröder än kvinnor i samma familj, 95% konfidensintervall 0.42 till 0.18 färre bröder. Analysen visar alltså ett signifikant mindre antal bröder bland män jämfört med kvinnor i samma familjer.

Det bör hållas i minnet att analysen är baserad på födelsetal mm år 2000, och är ogiltig i andra familjer än rena kärnfamiljer. Födelsetalen är olika i olika tider, men resultaten bör inte påverkas särskilt mycket av det. Den funna orättvisan i antalet bröder per syskon speglas av ett analogt fenomen som rör antalet systrar per syskon.

Ösimulering.

Den här ön har en höjdkarta som är helt deterministisk. Däremot är färgerna i varje ruta slumpmässigt vald, beroende på höjden. I appendix 3 ser du R-källkoden som ger dig en egen söderhavso med vulkan med snö på toppen. Lycklig resa.



Simulering av kartor I: småvägar och hus.

Nåt som är kul är att sitta en stund med en gammal karta och tänka sig att man letar upp ett sött litet torp med stockrosor och katt på trappan i solskenet. Som man åker till över veckoslutet, och bär ut en bricka med kaffe och sju sorters kakor till den vitmålade trädgårdsmöbeln. Ja, drömma kan man. Med programmet i appendix 1, anrop `ff(1)` respektive `ff(6)` simulerades följande kartor, som man kan ha som skrivbordsmönster om man vill drömma sig bort från kvartalsrapporten...

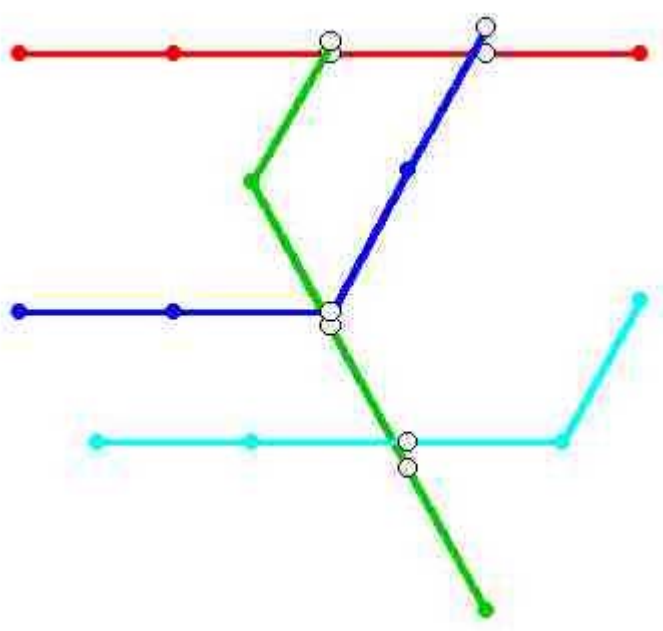


Miljöerna ser i de här två fallen ut som byar. Vägar till och från byarna saknas, men det går inte att avgöra om det är positivt eller negativt. Kartorna gjordes med programmet R version 1.8.1. [R Development Core Team (2003). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3, URL <http://www.R-project.org>]. Apropos detta projekt sa en medförfattare: "Bakom varje man står en oförstående kvinna".

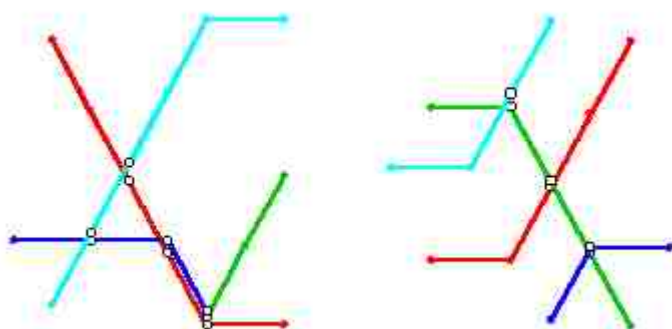
Simulering av kartor II: tunnelbanekartor.

Kartan över tunnelbanan i London lär ha blivit prisbelönt, för att på ett schematiskt sätt ha gjort det ganska komplexa bansystemet begripligt. Den är dock inte geografiskt korrekt, vilket är priset för att bli lättbegriplig. Tricket var visst, om jag minns rätt, att bara rita linjerna i vissa fasta vinklar, och med ungefär samma avstånd mellan intilliggande stationer. Detta gjorde kartan mindre rörig än verkligheten.

Jag tyckte detta var en kul idé, så jag simulerade kartor. Först beräknade jag ett antal tänkbara platser för stationer, och dessa platser var alla mittpunkter i ett bikakemönster. Det vill säga mittpunkter i sexkantiga celler. Endast tre riktningar är tillåtna, horisontellt, med 60 graders lutning, och med -60 graders lutning. Titta på exemplet för närmare definition av det där, när du ser det blir det mer begripligt än när jag förklarar det. Varje linje svänger i en viss (slumpmässigt vald) punkt, och svängen får inte vara för spetsig. Stationer med möjlighet att byta visas som ofyllda cirklar. Övriga stationer visas som fyllda cirklar. För att det inte ska bli rörigt är linjerna förskjutna i höjdlid. Ibland blir förskjutningen lyckad, ibland inte. Nedanstående figur visar ett relativt lyckat exempel.



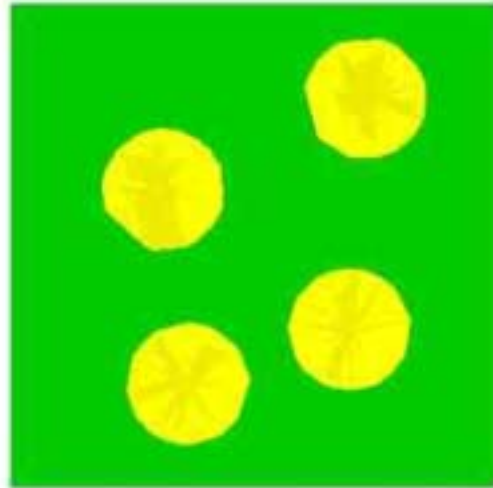
Genom att använda olika slumpvalsfrön får man förstås olika system. Här kommer två exempel till.



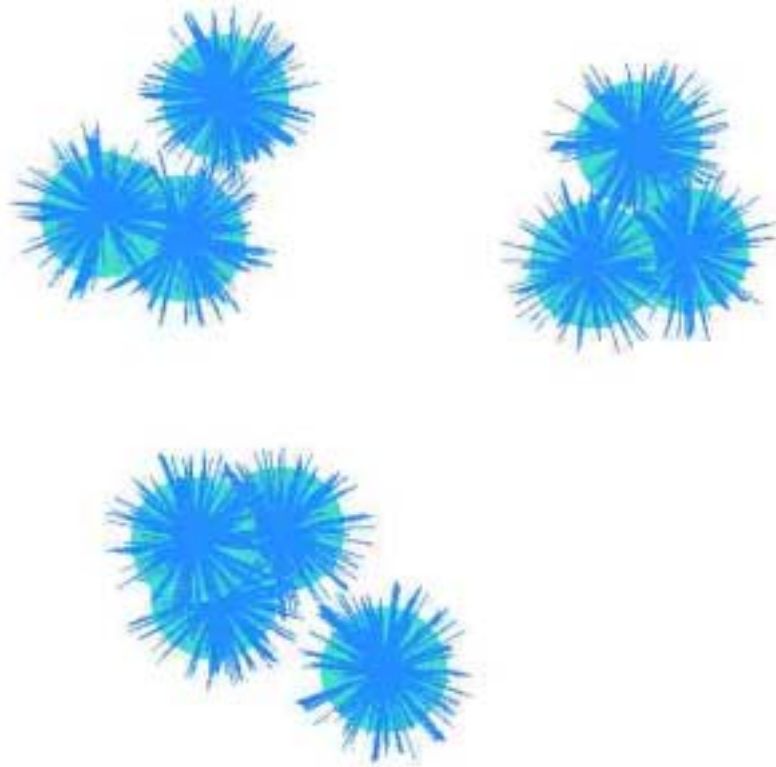
Det finns många sätt att bli gråhårig och detta är förvisso ett.

Datorgenererad bildkonst

Jag hackade nyligen ihop ett nöjesprogram som kan generera dekorativa mönster. Källkoden finns i bilaga 2. Programmet tar ett antal parametrar, och olika val kan ge väldigt olika resultat. De fyra mönsterna nedan är alla gjorda med detta program (källkoden visar vilka parametervärden som använts). Observera att programmet har körts med R version 1.8.1. När jag provade version 1.6.2 blev det helt andra mönster. Jag tror att slumpalsgeneratorn ändrats mellan dessa versioner.



Nedanstående bild har fått internationell spridning. Med internationell spridning menas att jag har skickat in den som bidrag i en tävling. Utfallet av tävlingen är ännu okänt. [Bilden genererades av: `bumlingar(2, c1="turquoise", c2="dodgerblue", m=10, n=250, R=0.1, R2=0.15, R1=0)`].



Appendix 1. Källkod till artikeln om simulering av kartor med småvägar och hus.

```
# q1.txt

rot <- function(vector, v)
{
  v <- v * pi/180
  vector %%% matrix(c(cos(v), sin(v), -sin(v), cos(v)), 2, 2, byrow=T)
}

l.p1.p2 <- function(p1, p2, ...)
  lines(x=c(p1[1], p2[1]), y=c(p1[2], p2[2]), ...)

len <- function(v)
  sqrt(sum(v^2))

no <- function(v)
  v/len(v)

# Konstanter
V <- 0.003
B <- 2
H <- 2
HT <- 15
J <- 2
K <- 2
W1 <- 0.25
W2 <- 0.75

grv <- function(p1, p2)
{
  v12 <- p2 - p1
  av12 <- rot(no(v12), 90) * V

  n <- len(v12) * HT

  l.p1.p2(p1+av12, p1+av12+v12, col="black")
  l.p1.p2(p1-av12, p1-av12+v12, col="black")
  for (i in 1:n)
  {
    s <- ifelse(i %%2 == 1, 1, -1)
    ph1 <- p1+runif(1)*v12+s*av12*B
    ph2 <- ph1 + s*av12*H
    ph3 <- ph2 + rot(av12*H, 90)
    ph4 <- ph3 - s*av12*H

    polygon(
      x=c(ph1[1], ph2[1], ph3[1], ph4[1], ph1[1]),
      y=c(ph1[2], ph2[2], ph3[2], ph4[2], ph1[2]),
      col="black", density=-1)
  }
}

qq <- function(p1, p2)
  p1 - rot(p2-p1, sample(c(0,1,-1, -15, 15, -30, 30, -45, 45, -90, 90), size=1)) *
  sample(c(1:3)/7, size=1)

zz <- function(p1, p2, tv.om=F)
{
  for (i in 1:sample(1:K, size=1))
  {
    p <- qq(p2, p1)
    grv(p2,p)

    if (len(p-p2) > 0.2)
    {
      zz(p, p2)
    }
  }

  if (!tv.om)
  {
    grv(p1,p2)
    zz(p2, p1, T)
  }
}

ff <- function(s)
```



```

{
  set.seed(s)
  par(mai=c(1,1,1,1)*0.1)

  plot(0,0,pch=32,xlim=c(-0.2, 1.2),ylim=c(-0.2, 1.2),
       xlab=" ", ylab=" ", axes=F)

  p1 <- runif(2)
  for (j in 1:J)
  {
    p <- runif(2)
    zz(p1, p)

    if (runif(1) < W1)
    {
      for (k in 1:J)
      {
        q <- runif(2)
        zz(p, q)

        if (runif(1) < W2)
        {
          for (d in 1:J)
          {
            z <- runif(2)
            zz(q, z)
          }
        }
      }
    }
  }
}

ff(6)
scan()
ff(9)

#EOF#

```

Appendix 2. Källkod till artikeln om datorgenererad bildkonst.

```

bumlingar <- function(
  s, m=10,n=10,
  R=0.25,R1=0.15,R2=0.10,
  c1=grey(0.7),c2=grey(0.85),c3="white")
{
  set.seed(s)
  par(mai=c(1,1,1,1)*0.1)
  plot(0,0,pch=32,xlim=c(-0.3,1.3), ylim=c(-0.3,1.3),xlab="",ylab="",axes=F)
  polygon(-0.3+1.6*c(0,0,1,1,0), -0.3+1.6*c(0,1,1,0,0),col=c3,border=c3)
  for (i in 1:m)
  {
    p <- runif(2)
    v <- sort(runif(n))*360
    v <- c(v,v[1])
    polygon(p[1]+cos(v*pi/180)*R, p[2]+sin(v*pi/180)*R, col=c1, density=-1, border=c1)

    r <- R1+runif(n)*R2
    r <- c(r,r[1])
    polygon(p[1]+cos(v*pi/180)*r, p[2]+sin(v*pi/180)*r, col=c2, density=-1, border=c2)
  }
}

par(mfrow=c(2,2))

bumlingar(1)
bumlingar(5,m=4,R=0.2,R1=0,R2=0.2,c1="yellow",c2="yellow2",c3="green3",n=30)
bumlingar(2,m=100, n=3, R1=0, R2=0.1, R=0.1, c1="purple", c2="blue", c3=grey(0.9))
bumlingar(6,m=10,n=20,R=0.1,R1=0,R2=0.2,c1="red1",c2="red4")

#EOF#

```

Appendix 3: källkod till artikeln om ösimulering.

k.txt - version 2.

```
dragö <- function()
{
  set.seed(1)

  x <- seq(-5, 5, by=0.1)
  y <- x
  n <- length(x)
  z <- matrix(0, n, n)

  for (i in 1:n)
    for (j in 1:n)
      z[i,j] <- 20/((1+x[i]^2+y[j]^2) +
                    3*(sin(x[i])+cos(y[j])) +
                    (-0.5)*(x[i]^2+y[j]^2) + 3.0

  z[z < -1] <- -1

  pc <- function(f) colours()[sample(f,1)]

  kol <- matrix("x", n-1, n-1)
  for (i in 1:(n-1))
  {
    for (j in 1:(n-1))
    {
      hörn <- z[i:(i+1),j:(j+1)]

      if (all(hörn < 0))
        kol[i,j] <- pc(c(562:566,123,43))
      else if (all(hörn > 15))
        kol[i,j] <- pc(c(223:253))
      else if (all(hörn > 10))
        kol[i,j] <- pc(c(161,183,187,197,205,215,220))
      else if (all(hörn > 2))
        kol[i,j] <- pc(c(86:88))
      else
        kol[i,j] <- pc(c(410:412))
    }
  }

  par(mai=c(0.1,0.1,0.1,0.1))
  persp(z, xlim=c(0.1, 0.9), ylim=c(0, 0.8),
        zlim=c(0,50), box=F, col=kol, border=NA)
}

#EOF#
```

Appendix 4: Program som gör det lite lättare att välja färger.

```
# färgväljare till diagram.

nm <- colours()

l <- length(nm)

h <- ceiling(sqrt(l)*2/3)
b <- ceiling(l/h)

print(c(b,h))

windows(6,4,6)
par(mai=c(0.1,0.1,0.1,0.1))

plot(0, 0, pch=32, xlim=c(0,1), ylim=c(0,1), axes=F,
     main="Click on box to identify colour; left of boxes to quit")

x <- rep(NA, l)
y <- rep(NA, l)

k <- 1
for (i in 1:h)
{
  for (j in 1:b)
  {
    if (k <= l)
    {
      x[k] <- (j-0.5)/b
      y[k] <- (i-0.5)/h
      points(x[k], y[k], col=nm[k], pch=15, cex=3)
      k <- k + 1
    }
  }
}

l <- list(x=1)
while (l$x > 0)
{
  l <- locator(1)

  f <- function(x, kand){which(abs(kand-x) == min(abs(kand-x)))}

  i <- f(l$x, x)
  j <- f(l$y, y)

  k <- i[i %in% j]

  cat(paste("Colour #", k, ", \"", nm[k], "\".\n"))
}

dev.off(dev.cur())

#EOF#
```